

PEMANFAATAN GOOGLE CALENDAR SEBAGAI SMS *ALERT*
DALAM SISTEM *MONITORING* JARINGAN KOMPUTER
(Studi Kasus : Jaringan Komputer Universitas Bengkulu)

SKRIPSI



OLEH
YODY FERDIANSYAH
G1A009039

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS BENGKULU
2014

**PEMANFAATAN GOOGLE CALENDAR SEBAGAI SMS ALERT
DALAM SISTEM *MONITORING* JARINGAN KOMPUTER
(Studi Kasus : Jaringan Komputer Universitas Bengkulu)**

SKRIPSI

**Diajukan Sebagai Salah Satu Syarat untuk Memperoleh Gelar Sarjana pada
Fakultas Teknik Program Studi Teknik Informatika
Universitas Bengkulu**



**OLEH
YODY FERDIANSYAH
G1A009039**

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS BENGKULU
2014**

MOTTO DAN PERSEMBAHAN

Motto :

“You will understand if you want to learn it.

You will learn if you want to try it.

You will try if you want to understand it”

(Yody Ferdiansyah, 2013)

Persembahan :

```
public void setDedication(People everyOne){  
    ▪ Allah SWT.  
    ▪ Nabi Muhammad SAW.  
  
    ▪ Keluarga besar (Alm) Ali Umar dan (Alm) Kisan, dan yang paling utama untuk ayah dan ibu tercinta, Yuherman dan Masyitah.  
    ▪ Saudara-saudara tersayang, Dedy Verdian, Edo Pratama, Harry Tsaputra dan Farhan Alfariqi.  
  
    ▪ Teman-teman, Randy, Handrie, Lian, Om Indro, Bang Diki, Bang Widi, Bang Ari, Azhar, Apni, Abdur, Abner, Bobby, Dian, Disa, Ejo, Eko, Ferry, Firdaus, Ghuftron, Indri, Linda, Ariansyah, Rinov, Munadi, Odie, Oni, Rozy, Ryza, Samuel, Tri, Wisnu, Yessica, Julia, Zulmi, Denny, Egi, Eleo, Fitri, Gita J., Dita, Aji, Irawan, Leni, Rewa, Meilia, Radas, Robbie, Rofika, Sostri, Yoggy, Mbak Anggun, Mbak Riza, Mbak Kimlin, Mbak Liga, Rizki, Gita T., Lisa, Atri, Ambar, Fathin, Dwinda, Nia, Dyan, Ance, Ayuk Fitri, Rati, Yolan, Leon, Nina, Misia, Cindy, Deta, Yesi, dll.  
    ▪ Teman-teman Teknik Informatika UNIB,  
  
    ▪ Teman-teman AIB, Alfizah, Andy, Meidy, Dita, Budi, Amelia, Agustawati, Adhiah, Ghaniy, Randi, Ryan, Arpian, Atik, Dede, Febza, Hartantyo, Henny, Purwantina, Sri, Sylvia, Sundari, Utri, Wirda, Yoga, Nuri, Adhetra, Tiara, Nolanda, Dhea, Rara, Inka, Habib, Naning, Bertha, Mifta, dll.  
  
    ▪ SD Negeri 20 Kota Bengkulu, SMP Negeri 4 Kota Bengkulu, SMA Negeri 2 Kota Bengkulu dan Almamater UNIB;  
}
```

KATA PENGANTAR



Assalamu'alaikum Warahmatullahi Wabarakatuh

Alhamdulillahirabbil'alamin, puji syukur penulis ucapkan kepada Allah SWT yang telah memberikan rahmat serta hidayah-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul **“Pemanfaatan Google Calendar Sebagai SMS Alert Dalam Sistem Monitoring Jaringan Komputer (Studi Kasus: Jaringan Komputer Universitas Bengkulu)”** dengan lancar.

Skripsi ini merupakan tugas akhir untuk memenuhi salah satu syarat dalam menyelesaikan studi di Fakultas Teknik Universitas Bengkulu. Selain itu, skripsi ini juga merupakan suatu pembelajaran bagi penulis dalam banyak hal, baik ilmu pengetahuan, praktik ilmu yang didapat selama kuliah di Teknik Informatika dan bekal ilmu ke depannya. Penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan, oleh karena itu penulis masih perlu banyak belajar lagi dan membutuhkan saran yang membangun dalam skripsi ini. Penulis juga menyadari bahwa tanpa bantuan, saran, dan bimbingan dari pihak-pihak yang telah membantu, penulis tidak akan mampu menyelesaikan tugas akhir ini. Untuk itu penulis mengucapkan terima kasih kepada yang terhormat:

1. Ayah (Yuherman), Ibu (Masyitah) dan saudara-saudaraku tersayang (Dedy Verdian, Edo Pratama, Harry Tsaputra, dan Farhan Alfariqi) yang selalu memberikan semangat dan dukungan dalam penyelesaian skripsi ini. Terutama ibu yang selalu mengingatkan penulis dan memberikan dukungan moril kepada penulis.
2. Bapak Khairul Amri, S.T., M.T. sebagai Dekan Fakultas Teknik Universitas Bengkulu yang telah memberikan kesempatan kepada penulis untuk menyelesaikan pendidikan di Fakultas Teknik Universitas Bengkulu.
3. Ibu Ernawati, S.T., M.Cs. selaku Pembimbing Utama yang penuh kesungguhan, kesabaran dan bersedia meluangkan waktu yang cukup

banyak untuk membimbing, mengarahkan dan memotivasi penulis dalam menyelesaikan skripsi ini.

4. Bapak Funny Farady C, S.Kom., M.T. selaku Pembimbing Pendamping yang selalu memberikan waktu, arahan dan saran kepada penulis sehingga penulis menjadi lebih termotivasi dan bersemangat untuk menyelesaikan skripsi ini.
5. Drs. Boko Susilo, M.Kom sebagai dosen Penguji Utama yang telah menyediakan waktu untuk mengarahkan dan memberikan masukan-masukan demi penyempurnaan skripsi ini.
6. Ibu Desi Andreswari, S.T., M.Cs., selaku ketua Prodi Teknik Informatika Universitas Bengkulu dan Penguji Pendamping yang telah menyediakan waktu untuk mengarahkan dan menguji skripsi ini.
7. Seluruh dosen dan staf karyawan, serta seluruh civitas akademika Program Studi Teknik Informatika Fakultas Teknik Universitas Bengkulu.
8. Mbak Fenty Adri sebagai staf administrasi Prodi Teknik Informatika yang telah membantu proses administrasi sehingga tugas akhir ini dapat selesai tepat waktu.
9. Bapak Bambang Aris, Amd., sebagai staf jaringan Lembaga Pengembangan Teknologi Informasi dan Komunikasi (LPTIK) Universitas Bengkulu yang telah memberikan izin penelitian dan memberi bantuan pengumpulan data untuk penyelesaian skripsi ini.
10. Teman-teman satu angkatan Teknik Informatika 2009 terutama Randy, Handrie, Nopiliansyah, Apni, Abdur dan Om Indro yang selalu memberikan saran dan dukungan.
11. Teman-teman satu SMA Negeri 2 Kota Bengkulu terutama Alfizah, Andy dan Meidy yang selalu memberikan saran dan dukungan.
12. Teman-teman satu jurusan Teknik Informatika baik kakak tingkat maupun adik tingkat yang selalu memberikan dukungan.
13. Semua pihak yang sudah membantu dan tidak dapat penulis sebutkan satu persatu.

Penulis berharap semoga Allah SWT mengaruniakan rahmat dan hidayah-Nya kepada mereka semua. Semoga skripsi ini dapat bermanfaat bagi kita semua, aamiin.

Wassalamu'alaikum Warahmatullahi Wabarakatuh

Bengkulu, Mei 2014

Penulis

PEMANFAATAN GOOGLE CALENDAR SEBAGAI SMS *ALERT*
DALAM SISTEM *MONITORING* JARINGAN KOMPUTER
(Studi Kasus : Jaringan Komputer Universitas Bengkulu)

Oleh
YODY FERDIANSYAH
NPM. G1A009039
E-mail : the.exact.boy@gmail.com

ABSTRAK

Jaringan komputer yang baik adalah jaringan yang selalu berada dalam keadaan stabil dan selalu siap melayani pengguna. Dalam menjaga kestabilan jaringan, dibutuhkan seorang administrator jaringan yang diharuskan untuk memantau keadaan jaringan secara terus menerus karena gangguan pada jaringan bisa terjadi kapan saja. Hal ini dapat berakibat fatal jika administrator jaringan sedang tidak berada di ruang *server* dan tidak mengetahui saat gangguan terjadi. Untuk mengatasi masalah ini, dirancang sebuah sistem *monitoring* jaringan berbasis Java Desktop dengan memanfaatkan notifikasi SMS dari Google Calendar. Metode pengembangan sistem yang digunakan untuk membangun aplikasi ini adalah model sekuensial linier. Sedangkan pada tahap analisa dan perancangan sistem dilakukan dengan pendekatan berorientasi objek menggunakan UML. Hasil akhir dari penelitian ini adalah sebuah sistem monitoring jaringan komputer yang mampu memberikan SMS *alert* sehingga administrator jaringan bisa mengetahui gangguan jaringan kapan saja dan dimana saja dan dapat menerima notifikasi jika terjadi gangguan jaringan.

Kata Kunci : Sistem Monitoring Jaringan, Google *Calendar*, SMS *Alert*, Java.

**USE GOOGLE CALENDAR AS SMS ALERT
IN A COMPUTER NETWORK MONITORING SYSTEM
(Case Study: Computer Networking in University of Bengkulu)**

By
YODY FERDIANSYAH
NPM. G1A009039
E-mail : the.exact.boy@gmail.com

ABSTRACT

A good computer network is a network that always in stable condition and be ready to serve users. For maintaining and make sure network in good status are needed a network administrator that must be ready to keep on eye on the network condition continuously, because the error network can be occurred in unpredictable time. It can be fatal if a network administrator is not in the network operation center (NOC) and don't know when the error network occurred. The solution to solve this problem is by designing the monitoring system or application to watch the network status by using Java Platform (desktop) and utilizing Short Message Service (SMS) notification from the service of Google Calendar. The system development method to designing this system is linier sequential model. Whereas, the system analysis and design phase is done by using object-oriented approached with Unified Modelling Language (UML). The result of this research is a completed system to safeguard or monitor the computer network status by sending a SMS alert to the network administrator so that can find out when the error network occurred in wherever they are, eventhough they are not in the NOC.

Keywords: Network Monitoring System, Google Calendar, SMS Alert, Java.

DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PERSETUJUAN.....	iii
LEMBAR PENGESAHAN SKRIPSI.....	iv
MOTTO DAN PERSEMBAHAN	v
KATA PENGANTAR	vi
ABSTRAK.....	ix
ABSTRACT.....	x
DAFTAR ISI.....	xi
DAFTAR GAMBAR	xiv
DAFTAR TABEL.....	xvii
DAFTAR LAMPIRAN.....	xviii
BAB I PENDAHULUAN	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	5
1.3. Batasan Masalah	5
1.4. Tujuan Penelitian.....	6
1.5. Manfaat Penulisan	6
BAB II LANDASAN TEORI	7
2.1. Pengertian Jaringan Komputer	7
2.1.1. Berdasarkan Area atau Skala	7
2.1.2. Berdasarkan Media Penghantar	8
2.1.3. Berdasarkan Fungsi.....	9
2.2. Pengertian TCP/IP	10
2.3. Pengalamatan IP	12
2.4. <i>Ping</i>	16
2.5. Sistem Monitoring Jaringan.....	16
2.6. Layanan Pesan Singkat.....	17
2.7. Karakteristik SMS	18
2.8. Google	19
2.8.1. Google APIs.....	19

2.8.2. Google Calendar.....	20
2.9. Tree.....	22
2.10. Netbeans IDE	24
2.11. Basis Data SQLite	25
2.12. <i>Unified Modelling Language</i> (UML).....	25
2.13. Metode Pengembangan Sistem	30
2.14. Pengujian Sistem.....	33
2.15. Penelitian Terkait	34
BAB III METODE PENELITIAN.....	36
3.1 Jenis Penelitian	36
3.2 Sarana Pendukung	36
3.3 Jenis dan Sumber Data	37
3.4 Teknik Pengumpulan Data	37
3.5 Metode Analisis Data	38
3.6 Metode Pengembangan Sistem.....	39
3.7 Metode Pengujian.....	43
3.8 Jadwal Penelitian	44
BAB IV ANALISIS DAN PERANCANGAN SISTEM	45
4.1 Kerangka Kerja Sistem.....	45
4.2 Identifikasi Masalah	46
4.3 Analisis Sistem	47
4.3.1. Pemahaman Kerja Sistem Yang Ada.....	47
4.4 Analisis Kebutuhan.....	49
4.4.1 Analisis Fungsional.....	49
4.4.2 Analisis Non-Fungsional	50
4.5 Perancangan Sistem.....	50
4.5.1 Perancangan Model UML (<i>Unified Modelling Language</i>).....	51
4.5.2 Perancangan Antarmuka	62
4.6 Perancangan Basis Data.....	69
4.7 Pengujian Sistem	70
BAB V HASIL PEMBAHASAN	71
5.1 Implementasi Antar Muka.....	71

5.2 Pengujian Sistem	72
5.2.1 Pengujian White Box	72
5.2.2 Pengujian Black Box	84
BAB VI KESIMPULAN.....	99
6.1 Kesimpulan.....	99
6.2 Saran	100
DAFTAR PUSTAKA	102
LAMPIRAN.....	104

DAFTAR GAMBAR

Gambar 2.1 Alamat IP Kelas A (Sofana, 2011).....	13
Gambar 2.2 Alamat IP kelas B (Sofana, 2011).....	13
Gambar 2.3 Alamat IP kelas C (Sofana, 2011).....	14
Gambar 2.4 Alamat IP kelas D (Sofana, 2011).....	15
Gambar 2.5 Alamat IP kelas E (Sofana, 2011).....	16
Gambar 2.6 Tree (Dwi, 2005).....	22
Gambar 2.7 Use Case Diagram (Pender, 2002).....	26
Gambar 2.8 Class Diagram (Pender, 2002).....	28
Gambar 2.9 Activity Diagram (Pender, 2002).....	29
Gambar 2.10 Sequence Diagram (Pender, 2002).....	30
Gambar 2.11 Model Sekuensial Linier (Pressman, 2002).....	31
Gambar 3.1 Diagram Alir Penelitian.....	40
Gambar 4.1 Kerangka Kerja Sistem.....	45
Gambar 4.2 Diagram Alir Sistem Monitoring Jaringan.....	47
Gambar 4.3 Use-Case Sistem Monitoring Jaringan.....	52
Gambar 4.4 Diagram Aktivitas Struktur Hierarki.....	53
Gambar 4.5 Diagram Aktivitas Proses <i>Ping</i>	54
Gambar 4.6 Diagram Aktivitas Pengiriman dan Penerimaan SMS.....	55
Gambar 4.7 Diagram Aktivitas Manajemen Perangkat.....	56
Gambar 4.8 Diagram Aktivitas Manajemen Akun Google.....	57
Gambar 4.9 Diagram Aktivitas Data Pesan Keluar.....	58
Gambar 4.10 Diagram Aktivitas Aktivasi Suara Peringatan.....	59
Gambar 4.11 Diagram Aktivitas Tentang Sistem.....	59
Gambar 4.12 Diagram Sekuensial Sistem Monitoring Jaringan.....	60
Gambar 4.13 Diagram Kelas Sistem Monitoring Jaringan.....	61
Gambar 4.14 Struktur Antarmuka Sistem Monitoring.....	62
Gambar 4.15 Desain Antarmuka Utama Sistem Monitoring.....	63
Gambar 4.16 Desain Pesan Konfirmasi Nonaktifasi Suara Peringatan.....	64
Gambar 4.17 Desain Antarmuka Data Pesan Keluar.....	64
Gambar 4.18 Desain Antarmuka Manajemen Perangkat.....	65

Gambar 4.19 Desain Antarmuka Tambah <i>Node</i>	66
Gambar 4.20 Desain Antarmuka Ubah <i>Node</i>	66
Gambar 4.21 Desain Antarmuka Hapus <i>Node</i>	67
Gambar 4.22 Desain Antarmuka <i>Properties</i>	67
Gambar 4.23 Desain Antarmuka Manajemen Akun Google	68
Gambar 4.24 Desain Pesan Validasi Akun Google	68
Gambar 4.25 Desain Antarmuka Tentang Sistem	69
Gambar 4.26 Desain Pesan Konfirmasi Keluar Sistem	69
Gambar 5.1 Tampilan Utama (Struktur Hierarki)	75
Gambar 5.2 Tampilan Utama (Pesan Sistem)	76
Gambar 5.3 <i>MenuItem</i> Pada Tampilan Utama	77
Gambar 5.4 Konfirmasi Nonaktif Suara Peringatan	77
Gambar 5.5 Konfirmasi Keluar Sistem	77
Gambar 5.6 <i>Form</i> Manajemen Perangkat	78
Gambar 5.7 Menu <i>Popup</i> Perangkat	79
Gambar 5.8 Menu Tambah <i>Node</i>	79
Gambar 5.9 Menu Ubah <i>Node</i>	80
Gambar 5.10 Menu Hapus <i>Node</i>	81
Gambar 5.11 Menu <i>Properties</i> /Detail Perangkat	81
Gambar 5.12 <i>Form</i> Akun Google	82
Gambar 5.13 Pesan Hasil Validasi	83
Gambar 5.14 <i>Form</i> Data Pesan Keluar	83
Gambar 5.15 <i>Form</i> Tentang Sistem	84
Gambar 5.16 Contoh Struktur Perangkat Jaringan UNIB	90
Gambar 5.17 Proses <i>Ping</i> Perangkat	90
Gambar 5.18 Pesan Status <i>Down</i>	91
Gambar 5.19 Menu suara peringatan	92
Gambar 5.20 Pemberitahuan membunyikan suara dan suara non-aktif	92
Gambar 5.21 Mengirim SMS Pemberitahuan	92
Gambar 5.22 Kegagalan Pengiriman SMS	93
Gambar 5.23 Sukses menyimpan data pesan keluar	93
Gambar 5.24 Gagal menyimpan data pesan keluar	93

Gambar 5.25 Perangkat berstatus <i>up</i>	94
Gambar 5.26 Pesan hasil <i>ping</i> sistem.....	94
Gambar 5.27 Data pesan keluar	94
Gambar 5.28 SMS yang diterima administrator	94
Gambar 5.29 Perangkat berstatus <i>Up</i>	95
Gambar 5.30 Perbedaan waktu pada Google Calendar.....	96
Gambar 5.31 Perangkat berstatus <i>down</i>	97
Gambar A.1 Kolom Pendaftaran Akun Google	1
Gambar A.2 <i>Popup</i> Pengantar Google Calendar	2
Gambar A.3 <i>Popup</i> Informasi Pengingat Acara	3
Gambar A.4 <i>Popup</i> Informasi Sinkronisasi Ponsel	3
Gambar A.5 Halaman Google Calendar	4
Gambar A.6 Menu Pengaturan Google Calendar	4
Gambar A.7 Pengaturan Sinkronisasi Seluler.....	5
Gambar A.8 <i>Popup</i> Pengingat dan Pemberitahuan.....	6
Gambar A.9 Tambah Pengingat	7
Gambar A.10 Jenis Pemberitahuan	7
Gambar A.11 Berbagi Kalender.....	8
Gambar A.12 Tambahkan Orang	8
Gambar A.13 Pemberitahuan Berbagi Kalender.....	9

DAFTAR TABEL

Tabel 2.1 IP <i>address private</i> (Sofana, 2011).....	12
Tabel 3.1 Jadwal Penelitian.....	44
Tabel 4.1 Kegiatan Aktor dan Interaksinya dengan Sistem.....	52
Tabel 4.2 Kelas, Atribut dan Method.....	62
Tabel 4.3 Tabel Perangkat.....	70
Tabel 4.4 Tabel Data Pesan Keluar.....	70
Tabel 5.1 Daftar Kelas dan Layout Aplikasi.....	71
Tabel 5.2 Pengujian Tampilan Menu Utama	85
Tabel 5.3 Pengujian Manajemen Perangkat.....	86
Tabel 5.4 Pengujian Tambah <i>Node</i>	87
Tabel 5.5 Pengujian Ubah <i>Node</i>	87
Tabel 5.6 Pengujian Hapus Node.....	87
Tabel 5.7 Pengujian Properties	88
Tabel 5.8 Pengujian Manajemen Akun Google	88
Tabel 5.9 Pengujian Data Pesan Keluar.....	89
Tabel 5.10 Pengujian Tentang Program.....	89
Tabel 5.11 Tabel Pengamatan Selisih Waktu Pengiriman dan Penerimaan SMS	95
Tabel 5.12 Perbandingan Jumlah <i>Event</i> dan Jumlah SMS.....	97

DAFTAR LAMPIRAN

Lampiran A-1 Tata Cara Pembuatan Akun Google	A-1
Lampiran A-2 Tata Cara Pengaktifan Google Calendar	A-2
Lampiran A-3 Tata Cara Sinkronisasi Perangkat Seluler	A-4
Lampiran A-4 Tata Cara Pengaturan Pengingat dan Pemberitahuan.....	A-6
Lampiran A-5 Tata Cara Berbagi Kalender Dengan Pengguna Lain	A-8
Lampiran B-1 Surat Permohonan Izin Penelitian	B-1
Lampiran B-7 Surat Keterangan Telah Melaksanakan Wawancara	B-2
Lampiran B-8 Surat Keterangan Telah Melaksanakan Pengambilan Data.....	B-3
Lampiran B-9 Surat Keterangan Telah Melaksanakan Pengujian Aplikasi.....	B-4
Lampiran C-1 Daftar Perangkat Jaringan Komputer Universitas Bengkulu.....	C-1
Lampiran C-2 Daftar <i>Event</i> Google Calendar (7 Mei 2014 – 20 Mei 2014).....	C-3
Lampiran C-3 Daftar SMS Pada Ponsel Pengguna (7 Mei 2014 – 20 Mei 2014).....	C-11

BAB I

PENDAHULUAN

1.1. Latar Belakang

Teknologi yang berkembang saat ini telah menjangkau ke segala lapisan masyarakat. Kebutuhan masyarakat akan teknologi informasi sudah tidak bisa dibendung lagi. Informasi bisa diperoleh masyarakat dengan mudah dengan menggunakan perangkat elektronik, misalnya komputer atau perangkat *mobile*. Pekerjaan yang dulunya dikerjakan secara manual, sekarang dapat mereka kerjakan secara otomatis sehingga pekerjaan mereka menjadi lebih efektif dan efisien. Terlebih lagi dengan jangkauan internet yang telah menyebar hingga ke pelosok pedesaan.

Menurut Kementerian Komunikasi dan Informasi pada tahun 2010 dalam tinjauannya mengenai distribusi komputer dan akses internet menyebutkan bahwa 67 persen distribusi komputer dan 70,05 persen akses internet tersebar di wilayah Jawa saja, sedangkan selebihnya berada di wilayah lain. Persentase kedua tertinggi ditempati oleh wilayah Sumatera dengan distribusi komputer mencapai 16,58 persen dan akses internet mencapai 16,77 persen. Selanjutnya, wilayah Kalimantan mencatatkan angka 5,91 persen dalam distribusi komputer dan 4,74 persen dalam akses internet (VIVAnews, 2010). Hal ini membuktikan bahwa kebutuhan masyarakat akan teknologi informasi, khususnya internet cukup tinggi di Indonesia. Kebutuhan akan teknologi informasi ini juga harus ditunjang dengan tersedianya jaringan internet yang baik dan mampu memenuhi kebutuhan masyarakat Indonesia.

Keberadaan jaringan internet secara tidak langsung membentuk suatu jaringan komputer yang saling terhubung dan membentuk suatu kesatuan. Jaringan komputer adalah kumpulan suatu gabungan berbagai perlengkapan komunikasi dan komputer yang dihubungkan satu sama lain melalui medium komunikasi secara elektronik. Medium komunikasi ini bisa berupa kabel untuk jenis LAN (*Local Area Network*) atau saluran telepon, gelombang mikro dan satelit hubungan WAN (*Wide Area Network*). Agar pertukaran informasi dapat berjalan dengan baik, maka perlu adanya kestabilan jaringan komputer itu sendiri.

Kestabilan jaringan adalah keadaan dimana suatu jaringan komputer memiliki tingkat keamanan yang kuat sehingga terjadi proses pertukaran data (koneksi) yang stabil, terstruktur, kuat serta mampu mengatasi berbagai gangguan. Logikanya, bila dalam suatu jaringan komputer ternyata memiliki sisi keamanan yang lemah tentu hal ini berdampak merusak kestabilan jaringan komputer tersebut. Banyaknya gangguan yang masuk akibat lemahnya keamanan dapat merusak kinerja transfer data dan merusak jaringan komputer itu sendiri.

Kerusakan jaringan komputer secara umum dapat berupa kerusakan perangkat keras (*hardware*) atau kerusakan konfigurasi perangkat tersebut. Berdasarkan gejala yang muncul, dapat ditentukan jenis kerusakan yang terjadi untuk dilakukan perbaikan secepatnya. Jika terjadi kerusakan pada perangkat jaringan, dapat dilakukan pengecekan konfigurasi maupun pengecekan fisik terhadap perangkat secara langsung. Untuk menjaga kestabilan jaringan dan menghindari kerusakan ini, setiap sistem memerlukan teknisi atau administrator jaringan.

Administrator merupakan bagian vital dari sebuah jaringan komputer. Administrator jaringan atau administrator sistem (*sysadmin*) adalah seseorang yang dipekerjakan untuk memelihara dan mengoperasikan sistem komputer dan / atau jaringan komputer. Tugas-tugas seorang administrator sistem sangat luas, dan sangat bervariasi dari satu organisasi ke yang lain. *Sysadmin* biasanya diisi dengan menginstal, mendukung, dan memelihara *server* atau sistem komputer lain, dan perencanaan untuk dan menanggapi pemadaman layanan dan masalah lainnya. Tugas lain mungkin termasuk *scripting* atau pemrograman ringan, manajemen proyek untuk sistem-proyek terkait, mengawasi pelatihan operator komputer dan menjadi konsultan untuk masalah komputer di luar pengetahuan staf dukungan teknis.

Google sebagai salah satu raksasa internet memiliki banyak fasilitas untuk penggunaannya. Salah satunya adalah Google Calendar. Fasilitas ini bermanfaat bagi pengguna untuk melakukan penjadwalan kegiatan menggunakan internet dan melakukan sinkronisasi dengan perangkat lainnya. Salah satu fitur yang dimiliki Google Calendar adalah pengingat (*reminder*). Fitur ini memungkinkan pengguna untuk mendapatkan pengingat untuk setiap kegiatan yang sudah dibuat di Google Calendar. Bentuk pengingat ini dapat berupa *e-mail*, *alert* (peringatan yang muncul di *browser*), dan pesan singkat (SMS). Penggunaan pengingat ini dapat disesuaikan oleh pengguna sesuai dengan kebutuhan.

Seiring berjalannya waktu, jaringan komputer yang memiliki tingkat keamanan tinggi sekalipun tidak bisa lepas dari gangguan. Gangguan yang terjadi biasanya disebabkan rusaknya perangkat secara fisik atau terjadinya perubahan konfigurasi dari perangkat tersebut. Untuk mencegah terjadinya gangguan, maka

administrator jaringan harus selalu berada di ruangan *server* untuk memantau keadaan jaringan. Tetapi kadang kala administrator jaringan tidak dapat selalu mengawasi koneksi jaringan secara 24 jam penuh. Keadaan ini juga terjadi pada jaringan komputer di Universitas Bengkulu dimana administrator jaringan tidak selalu berada di ruang *server* untuk melakukan *monitoring* dikarenakan adanya beberapa kegiatan dan tugas lainnya sebagai administrator jaringan.

Dari penelitian yang dilakukan oleh Taufan Dwi Prayogo dkk (2011), dibangun sebuah sistem yang memanfaatkan teknologi SMS *Gateway* yang mampu membantu administrator jaringan dalam melakukan monitoring jaringan kapan saja dan dimana saja untuk menjaga jaringan akan tetap stabil walaupun administrator jaringan sedang tidak di pusat pengawasan. Demikian halnya dengan penelitian yang dilakukan oleh Nurlela Dabukke dkk (2011), dibangun sebuah sistem monitoring yang mampu membantu administrator jaringan untuk mengetahui keadaan trafik jaringan yang diawasinya. Sistem ini menggunakan sistem SMS *Gateway* dan WAP *Gateway*. Dari hasil monitoring jaringan ini maka keadaan trafik jaringan bisa dikelompokkan menjadi tiga yaitu trafik dalam keadaan jaringan mati, trafik dalam keadaan jaringan sibuk dan trafik dalam keadaan jaringan normal. Sedangkan pada penelitian yang dilakukan oleh Mohamad Yasir (2010), dibangun sebuah sistem monitoring untuk membantu administrator dalam memantau keadaan jaringan menggunakan Gammu dan bahasa pemrograman Delphi 7. Dan yang terakhir, penelitian yang dilakukan oleh Rizqi Lutfia Chandra dkk (2011), dibangun sebuah sistem *monitoring* jaringan menggunakan metode pemrograman berorientasi objek, dan dilakukan penambahan modul untuk

pembacaan status dari nagios dan *script* untuk notifikasi sms. Proyek akhir dari penelitian ini dibangun dengan menggunakan bahasa Python.

Dari uraian diatas, penulis tertarik untuk merancang dan membangun sebuah sistem monitoring jaringan komputer dengan judul “Pemanfaatan Google Calendar Sebagai SMS *Alert* Dalam Sistem Monitoring Jaringan Komputer (Studi Kasus: Jaringan Komputer Universitas Bengkulu)”

1.2. Rumusan Masalah

Berdasarkan latar belakang di atas, rumusan masalah dalam penelitian ini adalah “bagaimana merancang dan membangun sistem monitoring jaringan yang memanfaatkan Google Calendar sebagai SMS *alert* (studi kasus : jaringan komputer Universitas Bengkulu) ?”

1.3. Batasan Masalah

Dalam penelitian ini, ditetapkan beberapa batasan sebagai berikut :

1. Masukkan dalam penelitian ini adalah perangkat jaringan yang memiliki IP statis dan terdaftar dalam jaringan komputer Universitas Bengkulu.
2. Proses yang dilakukan dalam penelitian ini adalah proses monitoring yang menghasilkan informasi berupa status perangkat (*up/down*) tanpa mendeteksi kerusakan fisik pada perangkat tersebut.
3. Keluaran yang dihasilkan dalam penelitian ini adalah
 - a) Pada sistem monitoring: susunan perangkat jaringan secara struktural, informasi daftar perangkat jaringan, *log* proses *Ping*, *log* proses pengiriman pesan, informasi suara peringatan, dan data pesan keluar (berdasarkan periode waktu tertentu).

- b) Pada sistem SMS *alert*: notifikasi dari Google Calendar mengenai perangkat yang mengalami perubahan status.

1.4. Tujuan Penelitian

Tujuan yang ingin dicapai dari penelitian ini adalah terciptanya sebuah sistem monitoring jaringan yang memiliki kemampuan dalam mengecek status perangkat jaringan setiap 5 menit dan jika terjadi perubahan status perangkat, sistem akan mengirimkan pemberitahuan berupa SMS dengan memanfaatkan fasilitas SMS *alert* dari Google Calendar.

1.5. Manfaat Penulisan

Adapun manfaat yang ingin dicapai dari penelitian ini adalah :

1. Membantu administrator jaringan komputer Universitas Bengkulu dalam memantau keadaan perangkat jaringan yang ada di Universitas Bengkulu.
2. Membantu memberikan notifikasi berbentuk SMS jika ada perangkat yang mengalami gangguan ketika administrator jaringan komputer Universitas Bengkulu sedang tidak di ruang *server*.

BAB II

LANDASAN TEORI

2.1. Pengertian Jaringan Komputer

Jaringan komputer (*computer networks*) adalah suatu himpunan interkoneksi sejumlah komputer *autonomous*. Dalam bahasa populer dapat dijelaskan bahwa jaringan komputer adalah kumpulan beberapa komputer (dan perangkat lain seperti *router*, *switch*, dan sebagainya) yang saling terhubung satu sama lain melalui media perantara. Media perantara ini bisa berupa media kabel ataupun media tanpa kabel (nirkabel). Informasi berupa data akan mengalir dari satu komputer ke komputer lainnya atau dari satu komputer ke perangkat lain, sehingga masing-masing komputer yang terhubung tersebut bisa saling bertukar data atau berbagi perangkat keras (Sofana, 2013).

Untuk memudahkan memahami jaringan komputer, para ahli membagi jaringan komputer berdasarkan beberapa klasifikasi, diantaranya:

2.1.1. Berdasarkan Area atau Skala

Menurut (Sofana, 2013) berdasarkan skala atau area, jaringan komputer dapat dibagi menjadi 4 jenis, yaitu:

1. LAN

Local Area Network adalah jaringan lokal yang dibuat pada area terbatas. Misalkan dalam satu gedung atau dalam satu ruangan. Kadangkala jaringan lokal disebut juga jaringan personal atau privat. LAN biasa digunakan pada sebuah jaringan kecil yang menggunakan *resource* secara bersamaan, seperti penggunaan *printer* secara bersama, dan sebagainya.

2. MAN

Metropolitan Area Network menggunakan metode yang sama dengan LAN namun daerah cakupannya lebih luas. Daerah cakupan MAN bisa satu RW, beberapa kantor yang berada dalam komplek yang sama, satu/beberapa desa, satu/beberapa kota. Dapat dikatakan MAN merupakan pengembangan dari LAN.

3. WAN

Wide Area Network cakupannya lebih luas daripada MAN. Cakupan WAN meliputi satu kawasan, satu negara, satu pulau, bahkan satu dunia. Metode yang digunakan WAN hampir sama dengan LAN dan MAN. Umumnya WAN dihubungkan dengan jaringan telepon digital. Namun media transmisi lain pun dapat digunakan.

4. Internet

Internet adalah interkoneksi jaringan komputer skala besar (mirip WAN), yang dihubungkan menggunakan protokol khusus. Jadi sebenarnya internet merupakan bagian dari WAN. Cakupan internet adalah satu dunia bahkan tidak menutup kemungkinan antarplanet. Koneksi antarjaringan komputer dapat dilakukan berkat dukungan protokol yang khas, yaitu TCP/IP (*Transmission Control Protocol/Internet Protocol*).

2.1.2. Berdasarkan Media Penghantar

Menurut (Sofana, 2013) berdasarkan media penghantar, jaringan komputer dapat dibagi menjadi 2 jenis, yaitu:

1. Wire Network

Wire network adalah jaringan komputer yang menggunakan kabel sebagai media penghantar. Jadi, data mengalir pada kabel. Kabel yang umum digunakan pada jaringan komputer biasanya berbahan dasar tembaga. Ada dua jenis kabel lain yang menggunakan bahan sejenis *fiber* yang disebut *fiber optic* atau serat optik. Biasanya bahan tembaga banyak digunakan pada LAN. Sedangkan untuk MAN atau WAN menggunakan gabungan kabel tembaga dan serat optik. Serat optik saat ini sudah semakin populer.

2. Wireless Network

Wireless network adalah jaringan tanpa kabel yang menggunakan media penghantar gelombang radio atau cahaya infrared atau LASER. Saat ini sudah semakin banyak *public area* atau lokasi tertentu yang menyediakan layanan wireless network. Sehingga pengguna dapat dengan mudah melakukan akses internet tanpa kabel. Frekuensi yang digunakan pada radio untuk jaringan komputer biasanya dikisaran 2.4 GHz dan 5.8 GHz. Sedangkan penggunaan *infrared* dan LASER umumnya hanya terbatas untuk jenis jaringan yang hanya melibatkan dua buah komputer saja atau disebut *point to point*.

2.1.3. Berdasarkan Fungsi

Menurut (Sofana, 2013) berdasarkan fungsi, jaringan komputer dapat dibagi menjadi 2 jenis, yaitu:

1. *Client Server*

Client Server adalah jaringan komputer yang mengharuskan salah satu (atau lebih) komputer difungsikan sebagai *server* atau *central*. Server melayani komputer lain disebut *client*. Layanan yang diberikan bisa berupa akses *web*, *e-mail*, file atau yang lain. *Client server* banyak dijumpai pada jaringan *internet*. Namun LAN atau jaringan lain bisa mengimplementasikan *client server*. Hal ini sangat bergantung pada kebutuhan masing-masing.

2. *Peer to Peer*

Peer to peer adalah jaringan komputer dimana setiap komputer bisa menjadis *server* sekaligus *client*. Jadi tidak ada komputer yang “lebih utama” dibandingkan dengan komputer lain. Setiap komputer dapat menerima dan memberikan *access* dari/ke komputer lain. *Peer to peer* banyak diimplementasikan pada MAN, WAN, atau internet. Namun hal ini kurang lazim. Salah satu alasannya masalah manajemen dan *security*. Cukup sulit menjamin *security* pada jaringan *peer to peer* manakala pengguna komputer sudah sangat banyak.

2.2. **Pengertian TCP/IP**

Protokol TCP/IP menyediakan sebuah metode pengalamatan bagi semua *host* yang terhubung dengan *network* TCP/IP. Alamat *host* ini disebut *IP address* misalnya 192.168.1.1. Alamat ini berbeda dengan *hardware address* misalnya kartu Ethernet atau *Network Interface Card* yang lain sehingga diperlukan konversi antara *hardware address* dengan *IP address* (Sofana, 2011).

IP *address* dibentuk oleh bilangan biner sepanjang 32bit, yang dibagi menjadi 4 bagian sehingga setiap bagian panjangnya 8 *bit*. IP *address* menjadi identifikasi setiap *node* pada jaringan internet. Artinya tidak boleh ada *node* lain (yang tergabung ke internet) menggunakan IP *address* yang sama. Contoh IP *address* sebagai berikut:

01000100 110000001 11111111 00000001

Apabila setiap bagian dikonversikan ke bilangan desimal maka IP *address* tersebut menjadi:

68.129.255.1

Bentuk penulisan IP *address* diatas dikenal dengan notasi “*doted decimal*”. Dalam praktiknya, IP *address* bentuk desimal inilah yang akan digunakan sebagai alamat *host*.

Lembaga yang mengatur alokasi IP *address* di berbagai negara adalah IANA (*Internet Assigned Numbers Authority*). Informasi tentang IANA dapat dijumpai pada situs <http://www.iana.org/abuse/>. Dalam praktiknya, pengguna internet tidak pernah berhubungan langsung dengan IANA. Sebagai gantinya, pengguna internet akan berhubungan dengan ISP setempat. ISP-lah yang akhirnya membagikan IP *address* kepada pengguna internet (Sofana, 2011).

IP *address* yang berjumlah sekitar 4 milyar ini tidak semuanya dapat digunakan sebagai IP *address* untuk *host*. Ada yang digunakan untuk keperluan khusus. Seperti untuk keperluan alamat *network*, alamat *broadcast*, alamat *localhost*, LAN, dan sebagainya. Menurut IANA, IP *address* berikut ini dicadangkan untuk keperluan jaringan intranet atau LAN (Sofana, 2011).

- a) Diawali dengan 10. (i.e. 10.0.0.0 hingga 10.255.255.255)

- b) Diawali dengan 127.
- c) Diawali dengan 169.254.
- d) Diawali dengan 172.16. hingga 172.31.
- e) Diawali dengan 192.168.

IP *address* yang digunakan untuk keperluan LAN/*intranet* disebut sebagai IP *address private*. Sedangkan IP *address* yang digunakan untuk keperluan internet disebut IP *address public*. Berikut ini daftar IP *address private*:

Tabel 2.1 IP *address private* (Sofana, 2011)

Kelas	IP address
A	10.0.0.0 – 10.255.255.255
B	172.16.0.0 – 172.31.255.255
C	192.168.0.0 – 192.168.255.255

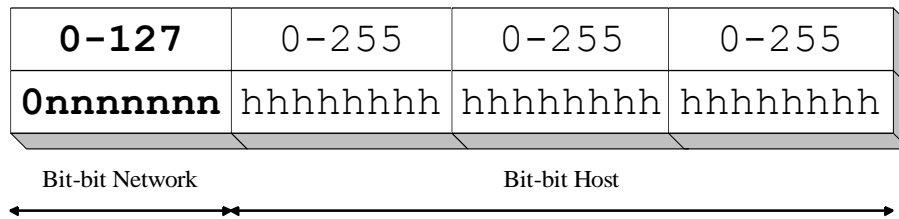
2.3. Pengalamatan IP

Jumlah IP *address* yang tersedia secara teoritis adalah 255x255x255x255 atau sekitar 4 milyar lebih yang harus dibagikan ke seluruh pengguna jaringan internet di seluruh dunia. Pembagian kelas-kelas ini ditujukan untuk mempermudah alokasi IP *address*, baik untuk *host*/jaringan tertentu atau untuk keperluan tertentu.

Untuk memudahkan pengaturan IP *address* pada seluruh komputer pengguna jaringan internet, dibentuklah suatu badan yang mengatur pembagian IP *address*. Badan tersebut bernama InterNIC (*Internet Network Information Center*). InterNIC bertugas untuk membagi IP *address* menjadi beberapa kelas. Kelas-kelas tersebut meliputi (Sofana, 2013):

1. Kelas A

Ip *address* kelas A memiliki struktur sebagai berikut:



Gambar 2.1 Alamat IP Kelas A (Sofana, 2011)

Jika *bit* pertama dari *IP address* adalah 0 maka *IP address* termasuk dalam *network* kelas A. *Bit* ini dan 7 *bit* berikutnya (8 *bit* pertama) merupakan *bit-bit network* (*network bit*) dan boleh bernilai berapa saja (kombinasi angka 1 dan 0), sedangkan 24 *bit* terakhir merupakan *bit host* (Sofana, 2013).

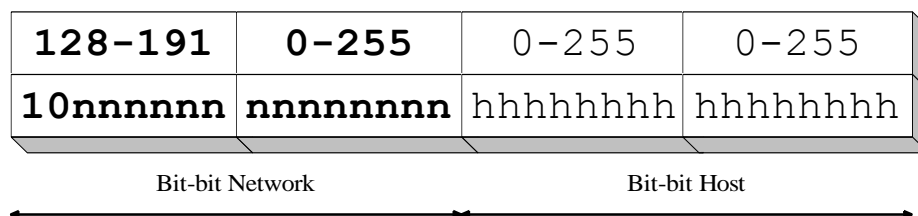
IP address kelas A dapat dituliskan sebagai berikut:

nnnnnnnn.hhhhhhhh.hhhhhhhh.hhhhhhhh

n menyatakan *network*, sedangkan *h* menyatakan *host*. Jangkauan *IP address* kelas A dimulai dari 1.xxx.xxx.xxx hingga 126.xxx.xxx.xxx. Setiap *network* dapat menampung lebih dari 16 juta (256^3) *host* (xxx adalah variabel, nilainya dari 0 hingga 255) (Sofana, 2013).

2. Kelas B

IP address kelas B memiliki struktur sebagai berikut:



Gambar 2.2 Alamat IP kelas B (Sofana, 2011)

Jika 2 *bit* pertama dari *IP address* adalah 10, maka *IP address* termasuk dalam *network* kelas B. Dua *bit* ini dan 14 *bit* berikutnya (16 *bit* pertama)

merupakan *bit network* dan boleh bernilai berapa saja (kombinasi angka 1 dan 0), sedangkan 16 *bit* terakhir merupakan *bit host*.

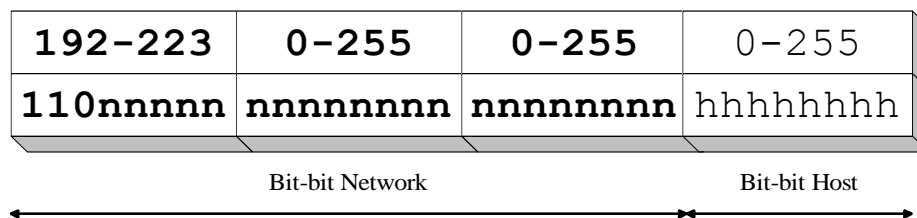
IP *address* kelas B dapat dituliskan sebagai berikut:

nnnnnnnn.nnnnnnnn.hhhhhhhh.hhhhhhhh

n menyatakan *network*, sedangkan *h* menyatakan *host*. Jangkauan IP *address* kelas B dimulai dari 128.0.xxx.xxx hingga 191.255.xxx.xxx. Setiap *network* dapat menampung lebih dari 65 ribu (256^2) *host* (Sofana, 2013).

3. Kelas C

IP *address* kelas C memiliki struktur sebagai berikut:



Gambar 2.3 Alamat IP kelas C (Sofana, 2011)

Jika 3 bit pertama dari IP *address* adalah 110, maka IP *address* dalam *network* kelas C. Tiga bit ini dan 21 bit berikutnya (24 bit pertama) merupakan *bit network* dan boleh bernilai berapa saja (kombinasi angka 1 dan 0), sedangkan 8 bit terakhir merupakan *bit host*.

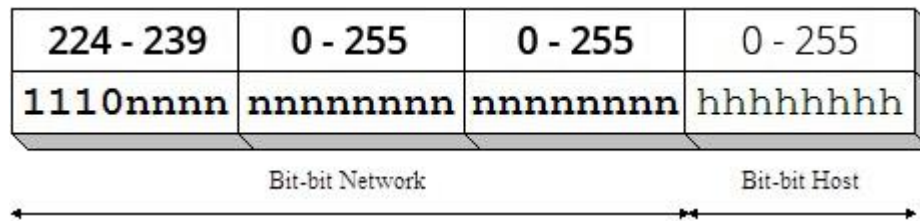
IP *address* kelas C dapat dituliskan sebagai berikut:

nnnnnnnn.nnnnnnnn.nnnnnnnn.hhhhhhhh

n menyatakan *network*, sedangkan *h* menyatakan *host*. Jangkauan IP *address* kelas C dimulai dari 192.0.0.xxx hingga 223.255.255.xxx. Setiap *network* dapat menampung sekitar 256 *host* (Sofana, 2013).

4. Kelas D

IP *address* kelas D memiliki struktur sebagai berikut:



Gambar 2.4 Alamat IP kelas D (Sofana, 2011)

Empat *bit* pertama bernilai 1110. IP *address* kelas D merupakan *multicast address*. Salah satu aplikasi yang memanfaatkan *multicast address* adalah *realtime video conferencing*. Pada IP *address* kelas D tidak dikenal *bit-bit network* dan *host* (Sofana, 2011).

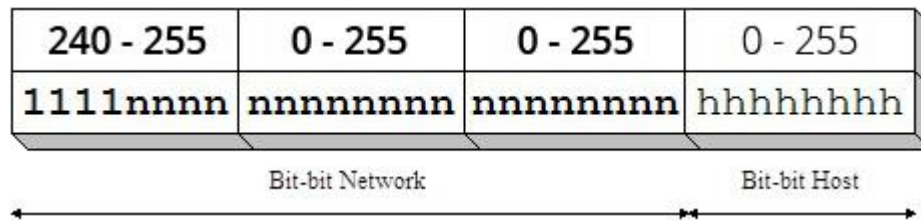
Pada jenis *traffic multicast*, sebuah paket yang dikirim ke alamat *multicast* akan diterima oleh semua *host* pada grup yang sama (alamat *multicast* yang sama). Identifikasi bagi setiap *host* pada grup tersebut cukup ditandai oleh sebuah IP *address* yang sama (Sofana, 2011).

Jangkauan IP *address* kelas D dimulai dari 224.0.0.0 hingga 239.255.255.255. IP *address* 224.0.0.0 hingga 224.0.0.255 dicadangkan untuk digunakan oleh protokol *network* pada sebuah segmen *network* lokal. Artinya, paket dengan alamat ini tidak akan di-*forward* oleh *router*. *Time-to-live* (TTL) paket akan diset 1 sehingga tidak akan di-*forward* oleh *router* (Sofana, 2011).

Sedangkan IP *address* dari 239.0.0.0 hingga 239.255.255.255 disebut *Administratively Scoped Address* yang dapat dianalogikan dengan *private address* 10.0.0./8 (kelas A) (Sofana, 2011).

5. Kelas E

IP *address* kelas E memiliki struktur sebagai berikut:



Gambar 2.5 Alamat IP kelas E (Sofana, 2011)

Empat *bit* pertama adalah 1111. IP *address* kelas E dicadangkan untuk kegiatan riset atau eksperimental. Pada IP *address* kelas E juga tidak dikenal *bit-bit network* dan *host* (Sofana, 2011).

2.4. Ping

Ping merupakan salah satu cara dasar untuk menguji koneksi perangkat dalam sebuah jaringan komputer. *Ping* yang berhasil akan ditunjukkan dengan terkirimnya paket yang dikirim dan diterima, tanpa *lost* tentunya. Selain digunakan antar media *peripheral* dalam jaringan komputer, *ping* juga dapat digunakan untuk mengecek koneksi jaringan komputer ke internet (Microsoft, 2013).

2.5. Sistem Monitoring Jaringan

Sistem monitoring jaringan menggambarkan sebuah sistem yang terus menerus memonitor jaringan komputer sehingga jika terjadi gangguan dapat secepatnya melakukan notifikasi kepada seorang *network administrator* atau sistem administrator. Sebagai contoh untuk mengetahui status dari sebuah *webserver*, *software monitoring* secara periodik mengirim *request http*; atau untuk *email server*, pesan tes di kirimkan melalui sebuah *SMTP* untuk kemudian di ambil melalui *IMAP* ataupun *POP3*. (Farhan, 2010)

Sistem monitoring jaringan merupakan bagian penting dalam sebuah sistem manajemen jaringan (*network management system*) sebagai upaya pencegahan insiden dengan memastikan bahwa *administrator* jaringan diberitahu ketika terjadi masalah pada jaringan. Walaupun *network monitoring* tidak dapat digunakan untuk menyelesaikan masalah ketika terjadi insiden, namun berbagai informasi penting dapat ditampilkan oleh sebuah aplikasi *network monitoring*.

Ada banyak *software* atau aplikasi monitoring jaringan, baik yang gratis maupun berbayar. Beberapa contoh aplikasi monitoring jaringan adalah Wireshark dan NetTools. Sedangkan aplikasi monitoring jaringan yang digunakan oleh universitas Bengkulu adalah Cacti.

2.6. Layanan Pesan Singkat

Layanan pesan singkat atau Surat masa singkat (*Short Message Service / SMS*) adalah sebuah layanan yang dilaksanakan dengan sebuah telepon genggam untuk mengirim atau menerima pesan-pesan pendek. Pada mulanya SMS dirancang sebagai bagian daripada GSM, tetapi sekarang sudah didapatkan pada jaringan bergerak lainnya termasuk jaringan UMTS dan CDMA.

Sebuah pesan SMS maksimal terdiri dari 140 *bytes*, dengan asumsi 1 *bytes* = 8-*bit*, maka sebuah SMS mengandung 1120-*bit* data. Dengan kata lain sebuah pesan bisa memuat 140 karakter 8-*bit*, 160 karakter 7-*bit* atau 70 karakter 16-*bit* untuk bahasa Jepang, Bahasa Mandarin dan Bahasa Korea yang memakai Hanzi (Aksara Kanji / Hanja). Adapula beberapa metode untuk mengirim pesan yang lebih dari 140 *bytes*, tetapi seorang pengguna harus membayar lebih dari sekali. SMS bisa pula untuk mengirim gambar, suara dan film. SMS bentuk ini disebut MMS (Yasir, 2010).

2.7. Karakteristik SMS

SMS *point-to-point* menyediakan mekanisme untuk mengirimkan pesan pendek (*short message*) ke dan dari piranti bergerak. Layanan ini menggunakan SMS Center (SMS *Center*) yang bertindak sebagai sistem simpan dan terusan (*store and forward*) untuk pesan pendek. (Baharuddin, 2008)

Menurut (Baharuddin, 2008) keberhasilan dan popularitas SMS antara lain disebabkan oleh:

1. Harga per kiriman tetap/ konstan

Apabila beban biaya telepon/ percakapan bervariasi, maka beban biaya kiriman SMS tetap.

2. Keamanan dan kesopanan

Apabila hendak menggunakan telepon seluler di tempat umum, maka berbicara saat menggunakannya dirasakan tidak sopan dan kurang aman. Namun sebaliknya berkirim pesan menggunakan SMS adalah lebih sopan dan *privacy* lebih terjaga.

3. Tidak mengganggu penerima

Seperti halnya *e-mail*, SMS sebagai alat komunikasi tidak mengganggu penerima, karena penerima bisa memutuskan kapan dan dimana dia akan menjawab pesan tersebut.

4. Handal (*reliable*)

Jaringan GSM secara umum diakui keandalannya dalam mengirimkan data dan SMS mewarisi kehandalan tersebut.

2.8. Google

Google adalah salah satu dari lima situs paling populer di dunia dan memungkinkan pengguna menemukan situs lain di web berdasarkan kata kunci pencarian. Google juga menyediakan pencarian khusus melalui *blog*, katalog, video, berita dan banyak lagi. Google bermula dari bertemunya Larry Page dan Sergey Brin di Stanford University dan berkolaborasi membangun mesin pencari yang disebut "BackRub". Nama ini berasal dari penggunaan mesin pencari *back-link* untuk menentukan relevansi halaman. Ini adalah algoritma yang dipatenkan dan dikenal sebagai *PageRank* (Karch(b), 2012).

Google menyediakan layanan Internet yang memungkinkan pengguna membuat *blog*, mengirim *email*, dan mempublikasikan halaman *web*. Google memiliki fasilitas jejaring sosial, *organizer*, dan aplikasi *chatting*, layanan untuk perangkat *mobile*, dan bahkan barang-barang bermerek Google. Layanan Google terbesar dan paling populer adalah pencarian web. Mesin pencari web Google dikenal untuk menyediakan hasil pencarian yang relevan dengan antarmuka yang bersih (Karch(b), 2012).

2.8.1. Google API

Google API adalah sebuah layanan dari Google yang memungkinkan pengembang untuk menemukan dan memanipulasi informasi di *web* secara lebih mudah. Fasilitas ini disediakan untuk pengembang dan peneliti yang tertarik untuk menggunakan Google sebagai sumber daya dalam aplikasi mereka. Dengan layanan ini, pengembang diijinkan untuk menggunakan lebih dari 3 milyar kueri dokumen di web secara langsung dari program komputer yang telah dibangun oleh pengembang (Raveendranathan, 2004).

Beberapa contoh produk Google yang dapat diakses menggunakan Google API adalah Google+ (Plus), Google Android, Google Chrome, Google Games, Google Maps, Google Apps, Google TV, Google Wallet dan YouTube. Untuk penggunaannya, pengembang *software* cukup mengunduh paket Google API pada halaman Google Developers sesuai bahasa pemrograman yang digunakan. Pengembang juga dapat menggunakan beberapa API sekaligus dalam satu sistem.

2.8.2. Google Calendar

Google *Calendar* adalah aplikasi pengelolaan waktu berbasis web dari Google dan memungkinkan pengguna melacak acara dan berbagi kalender dengan orang lain. Aplikasi ini diluncurkan pada 13 April 2006 dan keluar dari fase *beta* pada Juli 2009. Pengguna perlu memiliki akun Google untuk menggunakan perangkat lunak ini (Karch(a), 2012).

Google Calendar mengizinkan aplikasi yang dibuat oleh pengembang untuk melihat dan memperbarui data pada kalender menggunakan Google Data API. Pengembang juga dapat menggunakan Google Calendar Data API untuk membuat acara baru, mengubah atau menghapus acara yang ada, dan mencari acara dengan kueri yang cocok dengan kriteria tertentu pada aplikasi mereka. Pengembang bahkan dapat menentukan pengingat atau notifikasi saat membuat sebuah acara pada kalender. Bentuk pengingat ini dapat berupa *email*, *alert* (sebuah menu *popup* pada *browser*), atau SMS (layanan pesan singkat) (Google, 2013).

Meskipun pengaturan notifikasi SMS sudah diatur dengan benar, ada kalanya SMS notifikasi tidak diterima oleh pengguna. Hal ini disebabkan oleh adanya pembatasan jumlah SMS yang diterima pengguna dalam satu hari. Dengan pembatasan ini akan ada kemungkinan saat pengguna membuat *event* dalam jumlah

yang besar, tidak semua SMS notifikasi akan diterima oleh pengguna (Google(a), 2014).

Sebelum menerima notifikasi SMS dari Google Calendar, pengguna diharuskan untuk mengaktifkan fasilitas SMS yang dijelaskan pada lampiran A-3. Sedangkan cara untuk mendapatkan notifikasi SMS, pengguna diharuskan membuat *event* dengan menggunakan sistem pengingat berbentuk SMS sehingga saat waktu memulai *event* tercapai, maka Google Calendar akan mengirimkan notifikasi SMS yang berisi judul *event* tersebut. Berikut ini adalah cara pembuatan *event* pada Google Calendar API v2:

```
URL postUrl = new
URL("https://www.google.com/calendar/feeds/jo@gmail.com/private/full");
CalendarEventEntry myEntry = new CalendarEventEntry();

myEntry.setTitle(new PlainTextConstruct("Tennis with Beth"));
myEntry.setContent(new PlainTextConstruct("Meet for a quick lesson.));

DateTime startTime = DateTime.parseDateTime("2006-04-17T15:00:00-08:00");
DateTime endTime = DateTime.parseDateTime("2006-04-17T17:00:00-08:00");
When eventTimes = new When();
eventTimes.setStartTime(startTime);
eventTimes.setEndTime(endTime);
myEntry.addTime(eventTimes);

// Send the request and receive the response:
CalendarEventEntry insertedEntry = myService.insert(postUrl, myEntry);
```

Sedangkan untuk pengaturan notifikasi SMS pada *event* tersebut, ditambahkan sintaks berikut:

```
int reminderMinutes = 15;
Method methodType = Method.SMS;

Reminder reminder = new Reminder();
reminder.setMinutes(reminderMinutes);
reminder.setMethod(methodType);

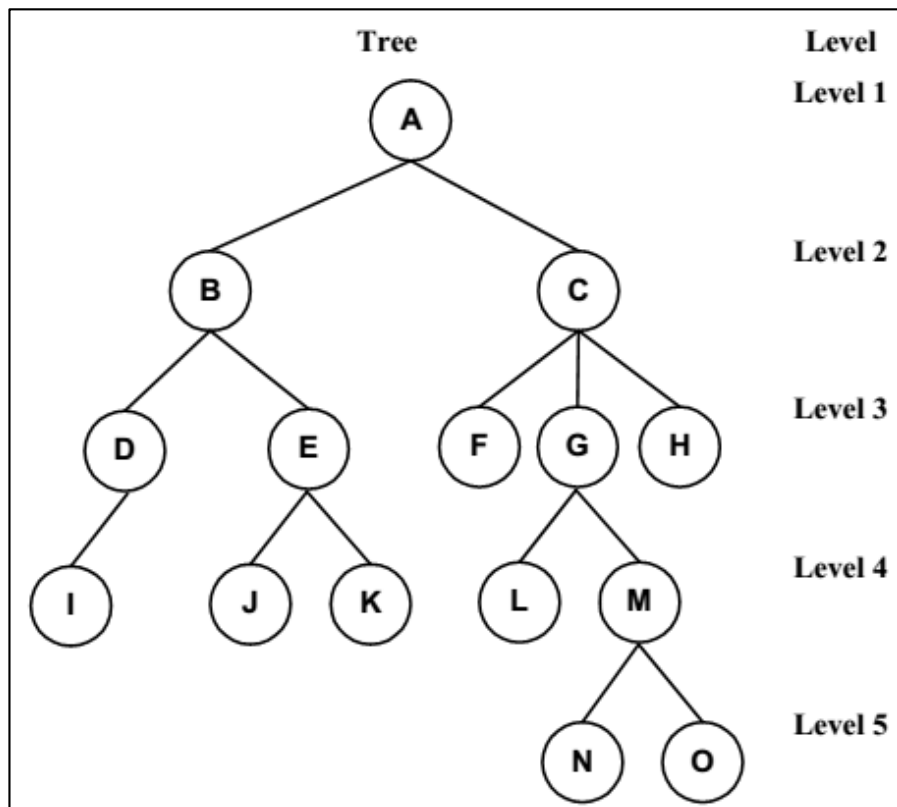
entry.getReminder().add(reminder);
entry.update();
```

Nantinya pengguna akan menerima notifikasi SMS 15 menit sebelum *event* dimulai.

2.9. Tree

Tree merupakan struktur data yang tidak linear/non linear yang digunakan terutama untuk merepresentasikan hubungan data yang bersifat hierarkis antara elemen-elemennya. Definisi *tree* secara lebih lengkap adalah kumpulan elemen yang salah satu elemennya disebut dengan *root* (akar) dan sisa elemen yang lain disebut sebagai simpul (*node/vertex*) yang terpecah menjadi sejumlah himpunan yang tidak saling berhubungan satu sama lain, yang disebut *subtree*/cabang (Dwi, 2005).

Ilustrasi dari *tree* dapat dilihat pada gambar 2.6 berikut ini:



Gambar 2.6 *Tree* (Dwi, 2005)

Struktur *tree* dalam penelitian ini akan digunakan untuk menampilkan susunan perangkat jaringan komputer universitas Bengkulu sesuai keadaan sebenarnya di lapangan.

Berikut ini adalah beberapa istilah objek *tree*:

1. Simpul adalah elemen *tree* yang berisi informasi/data dan penunjuk pencabangan.
2. Tingkat/level suatu simpul ditentukan dari akar (*root*), sebagai level 1. Apabila simpul dinyatakan sebagai tingkat N, maka simpul-simpul yang merupakan anaknya berada pada tingkat N+1.
3. Derajat/*degree* menyatakan banyaknya anak/turunan di simpul tersebut. Contoh: Simpul A memiliki derajat 2 (B dan C), simpul yang memiliki derajat 0 (nol) disebut *leaf* (daun) seperti: F, H, I, J, K, L, N, dan O.
4. Tinggi (*height*) atau kedalaman (*depth*) suatu *tree* adalah tingkat maksimum dari tingkat dalam *tree* tersebut dikurangi 1. Contoh dalam *tree* di atas, mempunyai *depth* 4.
5. *Ancestor* suatu simpul adalah semua simpul yang terletak dalam satu jalur dengan simpul tersebut, dari akar sampai simpul yang ditinjau. Contoh *Ancestor* L adalah A, C dan G.
6. *Predecessor* adalah simpul yang berada di atas simpul yang ditinjau. Contoh: *Predecessor* D adalah B.
7. *Successor* adalah simpul yang berada di bawah simpul yang ditinjau. Contoh: *Successor* D adalah I.
8. *Descendant* adalah seluruh simpul yang terletak sesudah simpul tertentu dan terletak pada jalur yang sama. Contoh: *Descendant* E adalah J dan K.
9. *Sibling* adalah simpul-simpul yang memiliki *parent* yang sama dengan simpul yang ditinjau. Contoh: *Sibling* J adalah K.

10. *Parent* adalah simpul yang berada satu level di atas simpul yang ditinjau.

Contoh: *Parent J* adalah E (Dwi, 2005).

2.10. Netbeans IDE

NetBeans merupakan sebuah proyek kode terbuka yang sukses dengan pengguna yang sangat luas, komunitas yang terus tumbuh, dan memiliki hampir 100 mitra kerja. Sun Microsystems mendirikan proyek kode terbuka NetBeans pada bulan Juni 2000 dan terus menjadi sponsor utama. NetBeans IDE adalah sebuah lingkungan pengembangan-sebuah perkakas untuk pemrogram menulis, mengompilasi, mencari kesalahan dan menyebarkan program. Netbeans IDE ditulis dalam Java - namun dapat mendukung bahasa pemrograman lain. Terdapat banyak modul untuk memperluas Netbeans IDE. Netbeans IDE adalah sebuah produk bebas dengan tanpa batasan bagaimana digunakan (Microsystems, 2000).

Netbeans IDE adalah aplikasi kode terbuka (*open source*) dan bebas (*free*) untuk penggunaan komersial dan non komersial. Kode sumber tersedia untuk guna ulang dengan lisensi *Common Development and Distribution License* (CDDL). (Microsystems, 2000)

Dalam penelitian ini, Netbeans digunakan untuk membangun tampilan dan proses yang ada pada sistem monitoring jaringan komputer. Tampilan yang digunakan berupa Java Swing, sedangkan proses yang dilakukan berupa proses ping, penyusunan tampilan perangkat berbentuk *tree* dan manajemen data. Untuk menampilkan susunan *tree* perangkat digunakan library *AbegoTreeLayout*, sedangkan untuk melakukan proses *ping* digunakan sintaks berikut:

```
InetAddress.getByName(ipAddress).isReachable(5000);
```


2.11. Basis Data SQLite

SQLite adalah perpustakaan basis data yang masih dalam proses pengembangan yang mengimplementasikan sistem berdiri sendiri, *serverless*, dan tanpa konfigurasi. Kode untuk SQLite berada dalam domain publik dan dengan demikian bebas untuk digunakan untuk tujuan apapun, komersial atau swasta. SQLite adalah mesin database SQL *embedded*. Tidak seperti kebanyakan database SQL lainnya, SQLite tidak memiliki proses *server* yang terpisah. SQLite membaca dan menulis secara langsung ke *filedisk* biasa.. Format *file database* adalah *cross-platform*-dapat dengan bebas menyalin *database* antara 32-bit dan 64-bit atau antara arsitektur *big-endian* dan *little-endian* (SQLite.org, 2014).

Penggunaan SQLite dalam bahasa pemrograman Java sangat mudah yakni dengan melakukan pengaturan koneksi dengan sintaks berikut ini:

```
Class.forName("org.sqlite.JDBC");  
Connection connection = DriverManager.getConnection  
("jdbc:sqlite:nama_database.db");
```

Sedangkan sintaks untuk proses pembuatan tabel dan proses manajemen data yang lain seperti *Create*, *Select*, *Insert*, *Update* dan *Delete* tetap sama seperti perpustakaan basis data yang lain misalnya MySQL dan SQL Server.

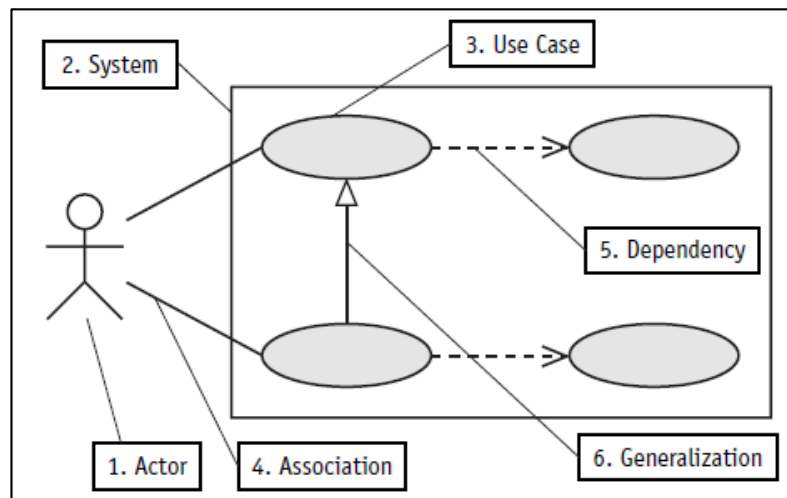
2.12. Unified Modelling Language (UML)

Unified Modelling Language (UML) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem perangkat lunak. Tidak semua diagram mutlak harus digunakan dalam pengembangan perangkat lunak, semuanya dibuat sesuai dengan kebutuhan (Widodo, 2011).

Berikut ini akan dijelaskan 4 diagram UML:

1. Diagram *Usecase*

Diagram *Usecase* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *usecase* merepresentasikan sebuah interaksi antara aktor dan sistem. *Usecase* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. Contoh diagram *use-case* dapat dilihat pada gambar 2.7 berikut ini.



Gambar 2.7 Use Case Diagram (Pender, 2002)

Ada 6 elemen yang membangun *use case* diagram: *actor*, *system*, *use cases*, *association*, *dependencies*, dan *generalization*.

1. *Actor*

Sebuah peran yang dimainkan oleh orang (misalnya pengguna sistem), sistem (misalnya “*Time*” atau sistem yang mengatur penjadwalan secara periodik atau bulanan), atau perangkat

(misalnya *Payment Processor*) yang memiliki saham dalam keberhasilan pengoperasian sistem.

2. *System*

Mengatur batas sistem dalam kaitannya dengan aktor yang menggunakannya (diluar sistem) dan fitur harus memberikan (dalam sistem).

3. *Use cases*

Mengidentifikasi fitur kunci dari sistem. Tanpa fitur ini, sistem tidak akan memenuhi pengguna / aktor persyaratan. Setiap *use case* mengungkapkan bahwa tujuan sistem harus tercapai.

4. *Association*

Mengidentifikasi interaksi antara aktor dan *use case*. Setiap asosiasi menjadi sebuah dialog yang harus dijelaskan dalam sebuah narasi *use case*. Setiap narasi dalam gilirannya memberikan seperangkat skenario yang berfungsi sebagai uji kasus ketika mengevaluasi analisis, desain, dan implementasi *use case*.

5. *Dependencies*

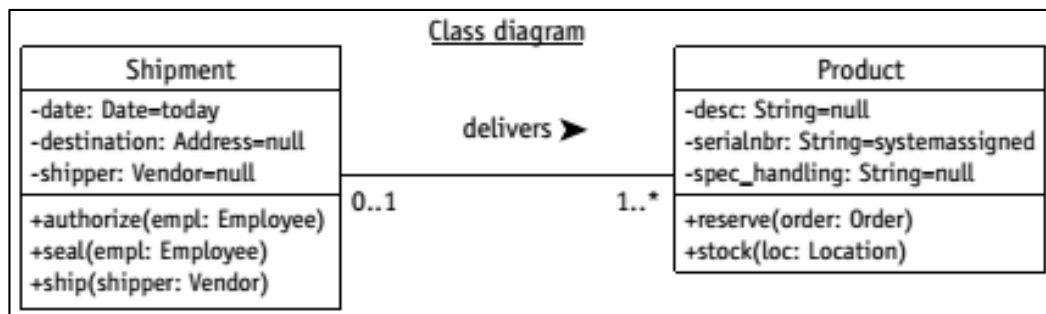
Mengidentifikasi hubungan komunikasi antara dua *use case*.

6. *Generalization*

Mendefinisikan sebuah hubungan antara dua aktor atau dua *use-case* yang mana salah satu *use-case* yang mewarisi dan menambah atau menggantikan sifat-sifat yang lain.

2. Diagram Kelas

Kelas adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. Kelas menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). Diagram kelas menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. Contoh diagram *class* dapat dilihat pada gambar 2.8 berikut ini.

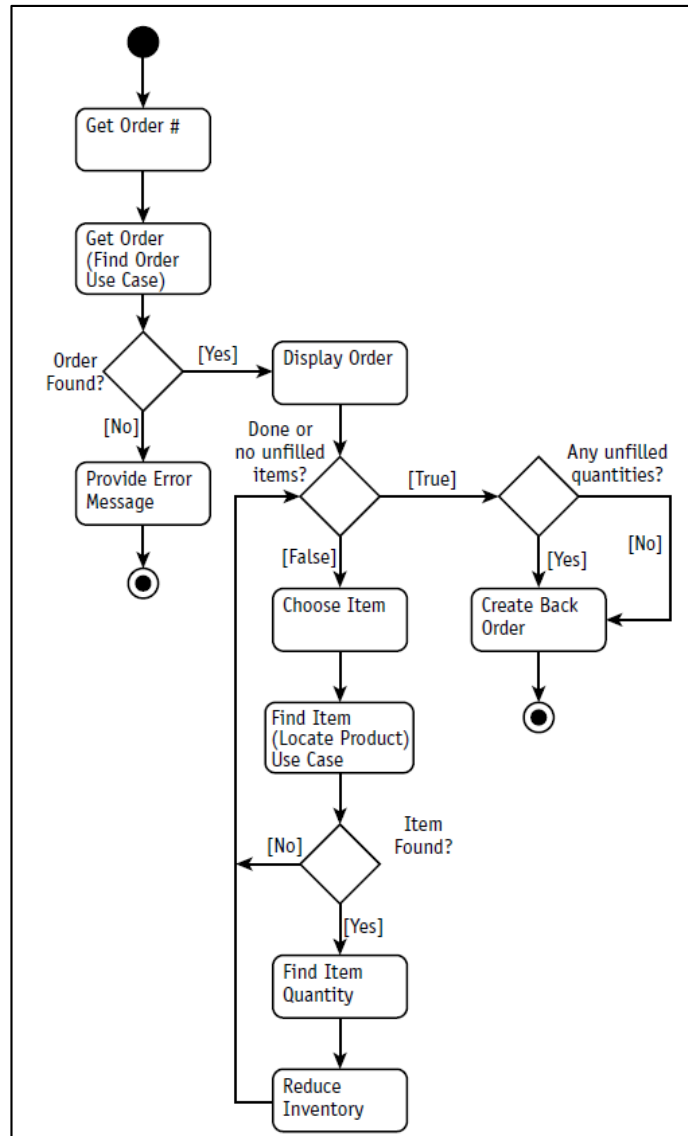


Gambar 2.8 Class Diagram (Pender, 2002)

3. Diagram Aktivitas

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity* diagram juga menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity* diagram merupakan state diagram khusus, dimana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh

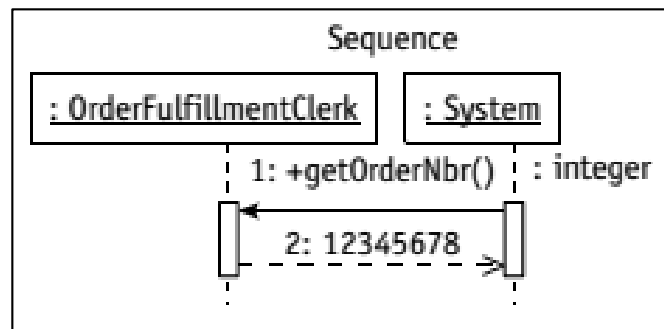
karena itu *activity* diagram tidak menawarkan *behavior internal* sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Contoh diagram *activity* dapat dilihat pada gambar 2.8 berikut ini.



Gambar 2.9 Activity Diagram (Pender, 2002)

4. Diagram Sekuensial

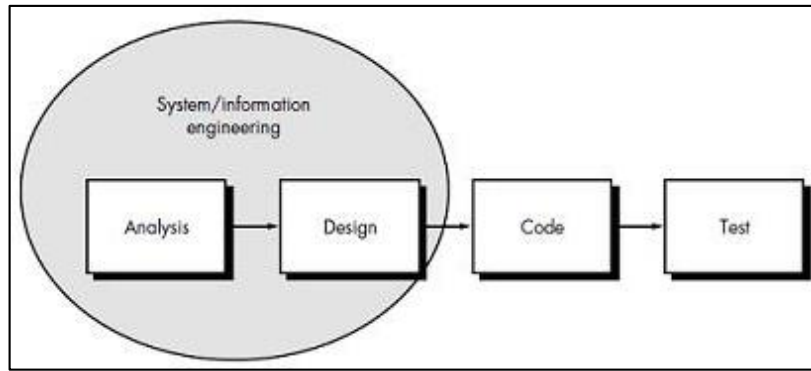
Diagram sekuensial menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah *event* untuk menghasilkan *output* tertentu. Contoh diagram *sequence* dapat dilihat pada gambar 2.9 berikut ini.



Gambar 2.10 Sequence Diagram (Pender, 2002)

2.13. Metode Pengembangan Sistem

Model sekuensial linier merupakan salah satu dari metode yang digunakan untuk pengembangan sistem. Sekuensial linier sering disebut juga dengan “siklus kehidupan klasik” atau “model air terjun”. Model sekuensial linier mengusulkan sebuah pendekatan kepada perkembangan perangkat lunak sistematis dan sekuensial yang mulai pada tingkat dan kemajuan sistem pada seluruh analisis, desain, kode, pengujian dan pemeliharaan (Pressman, 2002). Gambar 2.10 menggambarkan model pengembangan sistem sekuensial linier.



Gambar 2.11 Model Sekuensial Linier (Pressman, 2002)

Model sekuensial linier melingkupi aktivitas-aktivitas sebagai berikut (Pressman, 2002):

1. Rekayasa dan pemodelan sistem

Karena sistem merupakan bagian dari sebuah sistem yang lebih besar, kerja dimulai dengan membangun syarat dari semua elemen sistem dan mengalokasikan beberapa subset dari kebutuhan ke *software* tersebut. Pandangan sistem ini penting ketika *software* harus berhubungan dengan elemen-elemen yang lain seperti *software*, manusia, dan database. Rekayasa dan analisis sistem menyangkut pengumpulan kebutuhan pada tingkat sistem dengan sejumlah kecil analisis serta disain tingkat puncak. Rekayasa informasi mencakup juga pengumpulan kebutuhan pada tingkat bisnis strategis dan tingkat area bisnis.

2. Analisis kebutuhan *software*

Proses pengumpulan kebutuhan diintensifkan dan difokuskan, khususnya pada *software*. Untuk memahami sifat program yang dibangun, analisis harus memahami domain informasi, tingkah laku, unjuk kerja, dan interface yang diperlukan. Kebutuhan baik untuk sistem maupun *software* didokumentasikan dan dilihat lagi dengan pelanggan.

3. Desain

Desain *software* sebenarnya adalah proses multi langkah yang berfokus pada empat atribut sebuah program yang berbeda, struktur data, arsitektur *software*, representasi interface, dan detail (algoritma) prosedural. Proses desain menterjemahkan syarat/kebutuhan ke dalam sebuah representasi *software* yang dapat diperkirakan demi kualitas sebelum dimulai pemunculan kode. Sebagaimana persyaratan, desain didokumentasikan dan menjadi bagian dari konfigurasi *software*.

4. Generasi kode

Desain harus diterjemahkan kedalam bentuk mesin yang bisa dibaca. Langkah pembuatan kode melakukan tugas ini. Jika desain dilakukan dengan cara yang lengkap, pembuatan kode dapat diselesaikan secara mekanis.

5. Pengujian

Sekali program dibuat, pengujian program dimulai. Proses pengujian berfokus pada logika internal *software*, memastikan bahwa semua pernyataan sudah diuji, dan pada eksternal fungsional, yaitu mengarahkan pengujian untuk menemukan kesalahan-kesalahan dan memastikan bahwa *input* yang dibatasi akan memberikan hasil aktual yang sesuai dengan hasil yang dibutuhkan.

6. Pemeliharaan

Software akan mengalami perubahan setelah disampaikan kepada pelanggan (pengecualian yang mungkin adalah *software* yang dilekatkan). Perubahan akan terjadi karena kesalahan-kesalahan ditentukan, karena

software harus disesuaikan untuk mengakomodasi perubahan-perubahan di dalam lingkungan eksternalnya (contohnya perubahan yang dibutuhkan sebagai akibat dari perangkat peripheral atau sistem operasi yang baru), atau karena pelanggan membutuhkan perkembangan fungsional atau unjuk kerja. Pemeliharaan *software* mengaplikasikan lagi setiap fase program sebelumnya dan tidak membuat yang baru lagi.

2.14. Pengujian Sistem

Pengujian pada dasarnya adalah menemukan serta menghilangkan ‘*bug*’ (kesalahan-kesalahan) yang ada di sistem/perangkat lunak. Kesalahan-kesalahan itu dapat diakibatkan beberapa hal utama, antara lain kesalahan saat penentuan spesifikasi sistem, kesalahan saat melakukan analisis permasalahan, kesalahan saat perancangan, serta kesalahan saat implementasi (Nugroho, 2005).

Konsep kualitas sangat penting demi kepuasan pengguna (juga pengembang). Untuk mencapai kualitas yang diharapkan dari sistem yang akan dikembangkan pada umumnya ada beberapa strategi pengujian yang dapat dilakukan. Strategi-strategi itu adalah (Nugroho, 2005):

1. *Black-Box Testing*. Pada pengujian ini kita tidak perlu tahu apa sesungguhnya terjadi pada sistem/perangkat lunak. Yang kita uji adalah masukan serta keluarannya. Artinya, dengan berbagai masukan yang kita berikan, apakah sistem memberikan keluaran seperti yang kita harapkan?
2. *White-Box Testing*. Pengujian jenis ini mengasumsikan bahwa spesifikasi logika adalah penting dan perlu dilakukan pengujian untuk

menjamin apakah sistem berfungsi dengan baik. Tujuan utama dari strategi pengujian ini adalah pengujian berbasis kesalahan.

3. *Top-Down Testing*. Pengujian ini berasumsi bahwa logika utama atau interaksi antar objek perlu diuji lebih lanjut. Strategi ini seringkali dapat mendeteksi cacat/kesalahan/kekurangan yang serius. Pendekatan ini sesuai dengan strategi pengujian berbasis scenario.
4. *Bottom-Up Testing*. Strategi ini mulai dengan rincian sistem kemudian beranjak ke peringkat yang lebih tinggi. Dalam metodologi berorientasi objek, kita mulai dengan menguji metoda-metoda dalam kelas, menguji kelas-kelas serta interaksi antarkelas, dan selanjutnya hingga pada peringkat yang paling tinggi.

2.15. Penelitian Terkait

Hasil penelitian yang relevan dengan penelitian yang peneliti lakukan adalah:

1. Monitoring Jaringan *Wireless* Berbasis *SMS Gateway* dan *WAP Gateway* oleh Nurlela Dabukker, Reni Soelistjorini, dan Haryadi Amran Darwito dari Institut Teknologi Sepuluh Nopember pada tahun 2011. Penelitian ini menjabarkan tentang pembuatan sistem *SMS Gateway* dan *WAP Gateway* khusus pada *handphone* berbasis J2ME. Hasil dari penelitian ini adalah dapat diketahui keadaan trafik jaringan apakah mati, sibuk, atau normal. Sistem monitoring yang dihasilkan penelitian ini dapat memudahkan admin jaringan ketika sedang tidak didepan server karena dapat diaplikasikan ke *handphone* berbasis J2ME.

2. Sistem Monitoring Jaringan Pada Server Linux Dengan Menggunakan SMS *Gateway* oleh Taufan Dwi Prayogo, Kushartantya, dan Helmie Arif Wibawa dari Universitas Diponegoro pada tahun 2011. Penelitian ini menjabarkan tentang pembuatan sistem monitoring jaringan untuk membantu administrator jaringan mengawasi kondisi koneksi server linux.
3. SMS *Alert* Untuk *Reporting* Status Server oleh Rizqi Lutfia Chandra, Tribroto Harsono, ST., MT., dan Rolly Maulana Awangga, ST., dari Fakultas Informatika Institut Teknologi Telkom pada tahun 2011. Penelitian ini dapat memonitoring server dan membaca status per *service*-nya, kemudian di olah untuk notifikasi sms. Proyek akhir ini juga menyediakan aplikasi web untuk menampung *error* yang terjadi dan menyediakan fungsi untuk menginputkan saran perbaikan. Penelitian ini dibangun dengan menggunakan bahasa pemrograman Python.
4. Layanan Penyedia Informasi Kredit Ukm Berbasis SMS *Gateway* oleh Baharuddin dari Institut Teknologi Sepuluh Nopember Surabaya pada tahun 2005. Penelitian ini membangun sebuah server SMS *Gateway* yang bekerja tanpa adanya operator, mampu memberi informasi dan mencari informasi ke pihak perbankan sesuai dengan informasi yang diminta oleh pelanggan dan kemudian mengirimkan informasi tersebut kepada pelanggan yang meminta informasi tersebut.

BAB III

METODE PENELITIAN

3.1 Jenis Penelitian

Jenis penelitian ini adalah penelitian terapan (*applied research*), dimana biasanya penelitian ini dilakukan dengan mengambil permasalahan yang ada dalam sebuah organisasi atau perusahaan. Namun walaupun begitu, penelitian atau riset terapan ini memiliki nilai yang sama dengan riset dasar karena peneliti harus memiliki pengetahuan dasar dalam membangun kuesioner maupun faktor-faktor apa saja yang akan ditanyakan dan juga harus memiliki dasar dalam mengolah data secara statistik (Hasibuan, 2007).

Penelitian terapan ini bertujuan untuk merancang dan membangun sistem monitoring jaringan komputer Universitas Bengkulu dengan memanfaatkan Google Calendar sebagai SMS *alert*.

3.2 Sarana Pendukung

Sarana pendukung pembuatan sistem yaitu berupa *tool* dalam memperoleh data, pemodelan sistem hingga proses pembuatan sistem. Dalam penelitian ini sarana pendukung tersebut berupa perangkat keras (*hardware*) dan perangkat lunak (*software*) yang berperan penting dalam hal desain sistem dan *generate code* pada sistem.

1. Perangkat keras (*hardware*)

Adapun perangkat keras pada penelitian ini adalah: 1 unit Laptop dengan spesifikasi *Intel® Core™ i5-2450M CPU (2.50 GHz)*, RAM 4GB DDR3, HDD 500 GB, dan Printer *Canon PIXMA MP258*.

2. Perangkat lunak (*software*)

Adapun perangkat lunak pada penelitian ini adalah: Untuk rancang bangun perangkat lunak yaitu Sistem operasi *Windows Eight (8)* 32 bit, Bahasa Pemrograman *Java Development Kit (JDK) 7 Update 25*, Netbeans IDE 8.0, *Microsoft Office Visio 2013* untuk perancangan *flow-chart* dan *form* sistem, serta Astah Community untuk perancangan UML.

3.3 Jenis dan Sumber Data

Jenis data yang digunakan dalam penelitian ini adalah data sekunder berupa data-data perangkat yang terpasang dengan jumlah 80 data beserta deskripsi perangkat tersebut seperti jenis perangkat, lokasi, jumlah *port*, *port* yang digunakan, perangkat induk, *port* pada perangkat induk, susunan perangkat tersebut pada jaringan dan alamat IP yang digunakan oleh setiap perangkat. Data jenis perangkat pada penelitian ini terdiri dari *server*, *router*, *access point (wireless)*, *fingerprint* dan *switch*.

3.4 Teknik Pengumpulan Data

Dalam mengumpulkan data, teknik pengumpulan data yang digunakan dalam penelitian ini yaitu:

1. Wawancara

Wawancara dilakukan dengan staf jaringan LPTIK Universitas Bengkulu, bapak Bambang Aris, Amd. Dalam wawancara ini didapatkan data deskripsi perangkat jaringan beserta permasalahan yang terjadi pada jaringan komputer Universitas Bengkulu. Adapun sebagai lokasi wawancara adalah gedung LPTIK Universitas Bengkulu.

2. Dokumentasi

Diperoleh alamat IP perangkat jaringan yang ada pada sistem jaringan Lembaga Pengembangan Teknologi Informasi dan Komunikasi Universitas Bengkulu yang berjumlah 80 perangkat yang terdiri dari *server, router, switch, access point (wireless)* dan *fingerprint*.

3. Studi Kepustakaan

Studi pustaka dilakukan dengan mengumpulkan data dan informasi yang dijadikan sebagai acuan perancangan sistem monitoring jaringan komputer. Data dan informasi tersebut diantaranya: tutorial pemrograman jaringan menggunakan *java*, desain diagram UML, Google API, Google Calendar dan sumber-sumber tertulis baik tercetak maupun elektronik yang berhubungan dengan penelitian ini.

3.5 Metode Analisis Data

Jenis data yang digunakan dalam penelitian ini adalah data kualitatif yang berupa data perangkat jaringan. Setelah proses pengumpulan data, akan dilakukan analisis data sebagai berikut:

1. Format SMS yang akan diterima pengguna.

Sebelum mengirimkan notifikasi SMS, kita harus membuat format SMS yang informatif dan bersifat sederhana. Hal ini dikarenakan, untuk membuat notifikasi SMS dari Google Calendar, jumlah karakter maksimal yang dapat dimasukkan ke dalam teks SMS adalah 60 karakter. Berdasarkan keadaan diatas, maka format SMS yang akan digunakan adalah sebagai berikut:

Jenis_Perangkat/Alamat_IP/Deskripsi_Lokasi/Status/ (PID=ID Perangkat Induk/Port Pada Perangkat Induk)
--

Namun, dalam penerimaannya, format SMS ini akan ditambahkan kata-kata yang otomatis dibuat oleh Google. Penambahan kata ini berbeda untuk masing-masing jenis pengguna. Perbedaan bentuk SMS yang akan diterima oleh pengguna adalah sebagai berikut:

- a) Akun pengguna pemilik kalender

Pengingat: <FORMAT_SMS> - <waktu_pembuatan_event> (<alamat_email_pemilik>)

- b) Akun pengguna yang menerima pembagian kalender

<Nama_kalender> telah menambahkan <FORMAT_SMS> @ <waktu_pembuatan_event> (<alamat_email_pemilik>) pada kalender mereka
--

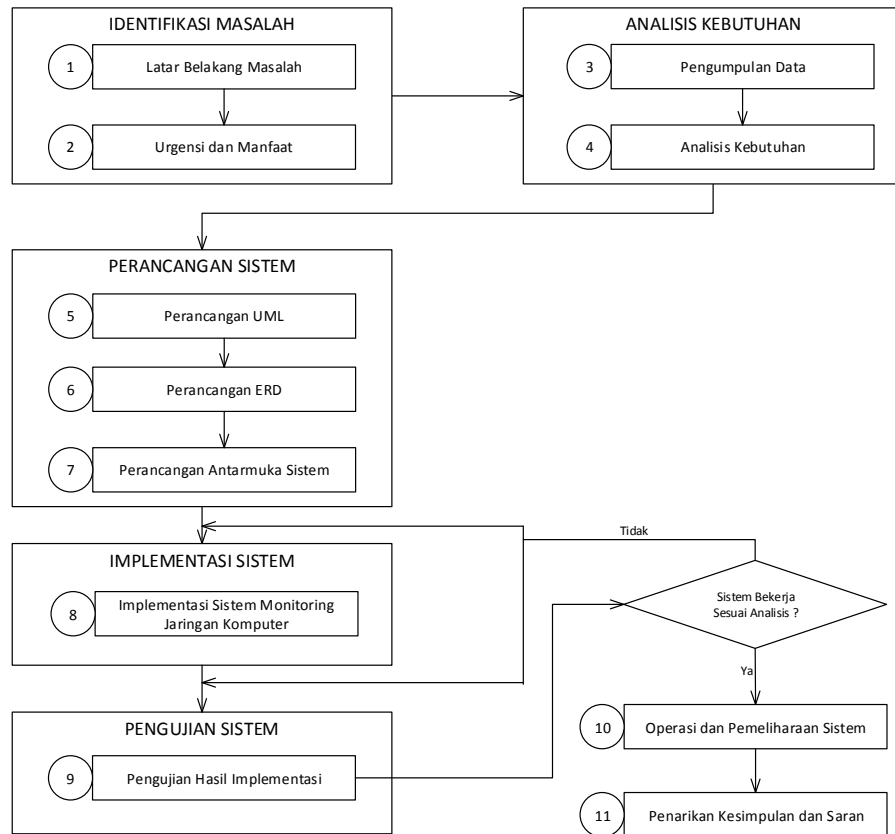
2. Kesesuaian proses pengiriman dan penerimaan SMS.

Untuk menentukan kesesuaian proses pengiriman dan penerimaan SMS, akan dilakukan beberapa pengamatan yaitu:

- a) Pengamatan untuk mengetahui kesesuaian antara tanggal pengiriman data *event* pada Google Calendar dan tanggal penerimaan SMS pada ponsel pengguna.
- b) Pengamatan untuk mengetahui sinkronisasi antara jumlah SMS yang dikirim oleh Google Calendar dengan jumlah SMS yang diterima oleh pengguna.

3.6 Metode Pengembangan Sistem

Metode pengembangan yang digunakan dalam penelitian ini adalah metode sekuensial linier yang ditunjukkan pada gambar 3.1 berikut ini:



Gambar 3.1 Diagram Alir Penelitian

Sistem yang dikembangkan dalam penelitian ini menggunakan model pengembangan sistem *sekuensial linier* seperti yang ditunjukkan pada gambar 3.1 disesuaikan dengan metode pengembangan sistem yang dipilih. Dari diagram tersebut, tahap-tahap yang akan dilakukan adalah

1. Pada tahap pertama adalah penentuan basis awal dari sebuah penelitian yaitu latar belakang penelitian.
2. Tahap kedua yaitu menentukan tujuan, manfaat dan ruang lingkup penelitian. Target pencapaian dalam tahap ini adalah diketahuinya tujuan dan manfaat dari pemanfaatan Google Calendar sebagai SMS *alert* dalam sistem monitoring jaringan komputer. Sedangkan batasan

masalah digunakan untuk membatasi pembahasan dan ruang lingkup penelitian agar tidak terlalu luas.

3. Pada tahap ketiga dilakukan pengumpulan data-data yang diperlukan dalam penelitian, meliputi jenis data masukan dan keluaran, spesifikasi fungsi, kemampuan dan fasilitas dari sistem monitoring jaringan komputer yang akan dibangun serta data-data deskripsi maupun abstrak hasil penelitian. Pengumpulan data tersebut akan diperoleh dari proses dokumentasi dan wawancara di Lembaga Pengembangan Teknologi Informasi dan Komunikasi (LPTIK) Universitas Bengkulu.
4. Tahap keempat adalah proses analisis kebutuhan. Analisis kebutuhan yang ditentukan oleh peneliti terdiri dari analisis kebutuhan sistem, analisis kebutuhan proses, analisis kebutuhan input, analisis kebutuhan output, dan analisis kebutuhan perangkat lunak dan perangkat keras.
5. Tahap kelima yaitu perancangan *Unified Modelling Language* (UML). Pada tahapan ini akan diketahui semua entitas luar, input, dan output yang terlibat dalam sistem serta diagram *use-case*, diagram kelas, diagram aktifitas, dan diagram *sequence* yang digunakan dalam analisis sistem.
6. Tahap keenam yaitu perancangan Entity Relationship Diagram (ERD). Perancangan meliputi beberapa proses yaitu pembuatan tabel *database* dan perancangan relasi antar basis data. Hasil dari tahap ini dapat dijadikan sebagai dasar dari implementasi basis data sistem nantinya.

7. Tahap ketujuh yaitu perancangan *flowchart* dan antarmuka (*human interface*). Tahap perancangan *flowchart* akan digunakan untuk menggambarkan sistem baru yang akan dikembangkan secara logis tanpa mempertimbangkan terlebih dahulu lingkungan sistem. Tahap perancangan antarmuka akan dibuat dalam beberapa rancangan tata letak sistem sesuai dengan analisis kebutuhan dari sistem.
8. Tahapan kedelapan yaitu implementasi sistem secara keseluruhan. Implementasi berdasarkan hasil dari tahapan keempat hingga ketujuh, yaitu dari tahapan analisis kebutuhan, UML, ERD, *flowchart* dan antarmuka. Pada tahap ini juga dilakukan pengkodean menggunakan Google API pada sistem.
9. Tahapan kesembilan yaitu tahapan pengujian hasil implementasi sistem secara umum. Pengujian meliputi pengujian input, proses dan output sistem apakah telah sesuai dengan perancangan sistem sebelumnya. Pada tahapan ini juga dilakukan pengujian pemanfaatan Google Calendar sebagai SMS *alert*. Tahapan ini menguji implementasi tahapan monitoring jaringan, pembaruan status perangkat dan pengiriman SMS menggunakan Google Calendar. Proses pengujian ini dilakukan menggunakan metode *black-box testing* dan *white-box testing*.
10. Tahapan kesebelas adalah operasi dan pemeliharaan sistem. Tahapan ini dilakukan setelah proses pengujian sistem telah berhasil dilakukan. Pengoperasian dilakukan langsung oleh pengguna sistem di Lembaga

Pengembangan Teknologi Informasi dan Komunikasi (LPTIK)
Universitas Bengkulu.

11. Tahapan terakhir adalah penarikan kesimpulan dan saran. Tahap tersebut dilakukan untuk mengetahui hasil yang telah diperoleh selama melakukan penelitian.

3.7 Metode Pengujian

Metode pengujian sistem yang digunakan dalam penelitian ini adalah *black-box testing* dan *white-box testing*. Hal yang akan diuji pada pengujian *black-box* antara lain:

1. Pengujian fungsional sistem dari sistem monitoring jaringan komputer.
2. Pengujian proses *ping* pada sistem monitoring jaringan komputer.
3. Pengujian untuk mengamati selisih waktu antara pengiriman dan penerimaan SMS.
4. Pengujian untuk mengamati sinkronisasi antara jumlah SMS yang dikirim dan jumlah SMS yang diterima.

Sedangkan hal yang akan diuji pada pengujian *white-box testing* antara lain:

1. Pengujian *method* dan *attribute* sistem monitoring jaringan komputer.
2. Pengujian antarmuka sistem monitoring jaringan komputer.

3.8 Jadwal Penelitian

Tabel 3.1 Jadwal Penelitian

No	Kegiatan	Des-13				Jan-14				Feb-14				Mar-14				Apr -14				Mei-14				Jun-14			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1.	Identifikasi Masalah																												
	Latar belakang penelitian																												
	Tujuan dan ruang lingkup penelitian																												
2.	Definisi kebutuhan																												
	Pengumpulan Data																												
	Analisis data yang dibutuhkan																												
4.	Perancangan sistem																												
5.	<i>Coding</i>																												
6.	Pengujian dan evaluasi																												
7.	Analisis hasil																												
8.	Pembuatan laporan																												